Program 2
50 points

Assigned:	Sept 11, 2014
Notification of team name and team leader:	Sept 15, 2014
Design approval due:	Sept 19, 2014
Implementation due:	Sept 25, 2014

Assignment: Using object-oriented programming techniques, design and implement a card game of your choosing. A `Card` class is provided for you to download from MyGCC. Some games to consider include *Black Jack*, *War*, and *Hearts*, among others. You will work in groups of your own choosing, with between three and four students in each group.

Notification. You must notify me by September 15 concerning your team members and your choice of a team leader. You may do so via an email from your team leader.

This group project will proceed in two phases: the *design phase* and the *implementation phase*. Each phase is outlined below.

Design phase. The focus of this assignment is on object-oriented program design. As such, you must think carefully about how C++ classes can be used to represent objects from the problem domain (in this case, a card game) as well as the interactions among them. You should consider the game elements, the rules, and so forth, to determine a collection of user-defined objects that can be used to implement the game.

When your design is complete, schedule an appointment with the instructor to present your design and obtain approval for it. **Design approval must be obtained by class time on Friday, September 19, 2014.** Once your design is approved, you will be permitted to continue with the implementation phase, which is described below.

Observe that the design can be made harder or easier by the complexity of the game rules, as well as by the complexity of the artificial intelligence that may be required. Whereas a certain minimum complexity is necessary to demonstrate the power of the object-oriented programming paradigm, choose your game wisely: a solid design is more important than a sophisticated implementation for this assignment.

Implementation phase. In this phase, you will actually implement the program design that was approved in the previous phase. If you have thought carefully about the design of your program, the necessary coding tasks should be straightforward. **The implementation is due before 11:55 PM on Thursday, September 25, 2013.**

Note that, in addition to the `Card` class provided with this document, your solution must utilize at least two other user-defined classes in a manner appropriate to the game of your choosing. (It is likely, however, that your solution will require more than just two such classes.)

Other requirements. You will work in groups of your own choosing with not more than four members per group to complete this assignment. You must nominate one member to be the "group leader"; this student will be responsible for submitting the solution on behalf of all the group members. **You must inform the instructor concerning who your group members are and who your group leader is by 5:00 PM on Monday, 15 September 2013.** A simple e-mail message will suffice.

You must produce a significant amount of code, you must involve all your team members, and **you must use a version control system to coordinate your work**. Most students prefer to work with Github (Github.com), which provides a free account in exchange for your making your source code available to its members. You can still achieve a secure version control system, however, by restricting to your team the rights to update the code. If you desire a private on-campus solution, a Subversion server will be made available to you. If you have not used version control, please view the excellent introduction by Software Carpentry at http://www.youtube.com/watch?v=fp-yJvZtWxE.

DO NOT REUSE EXISTING CODE for this. Your code should be original with your team.

Each group will have an opportunity to present its solution to the class on Friday, 26 September 2014. Presentations will proceed first by selecting volunteers and then by selecting groups at random if no volunteers are forthcoming. Note that the presentation can be informal (that is, you are not required to create PowerPoint slides or the like), but it must be informative.

Note that although this project encourages and requires collaboration within each group, the Honesty in Learning and Plagiarism Policy, as outlined in the course syllabus and in the 2014-2015 Crimson, still applies across group boundaries.

Rubric: To receive credit, your team leader must submit via MyGCC a `.zip` file containing the following items by 11:55 PM on the due date listed above:

1. **Source code (45 points):** Submit your entire Visual Studio project, including the source code and any non-standard headers or source files on which your program depends.

   Your code must compile, link, and execute using Visual Studio 2012: no compile equals no credit.

   Modularity is required in the design of your program. Use good software design and engineering techniques, such as procedural abstraction, object-oriented programming, and separation of interfaces/implementations (as appropriate), to increase the extensibility, maintainability, and readability of your code.

   In addition, you must include a comments section at the beginning of each of your source code files that provides a description of the code and its intended purpose. For example,

   ```
   /*
     Author:  Dorian P. Yeager
     Course:  COMP 220, Computer Programming II
     Date:    30 August 2014
     Description:  This file implements the driver
                   program for this assignment.
   */
   ```

   The following criteria will be used to grade your submission:

   - Does the code compile?
   - Does the code function according to the problem specification?
   - Is there an appropriate comments section at the beginning of each file (similar to the one shown above)?
   - Is the code readable and well-formatted? Is it well-documented and clear?

   The available points are distributed as follows:

Correctness:	35 points	*Supports course outcomes 4 & 5*
Comments:	5 points	
Organization:	5 points	

   A program that does not compile or link will not be graded.

2. **Supplemental information (5 points):** A text file with answers to the following questions:

   (a) How long did you spend on each part of this assignment?

   (b) Was the assignment difficult for you? Why or why not?

   (c) Did you enjoy working as a group to solve this problem? Why or why not?

   Feel free to expound or to be brief. You will not be graded on the responses themselves, only on the presence or absence of the responses. Your answers will be used to help improve this assignment in future incarnations of the class.

Submission: Programs must be submitted electronically via MyGCC before 11:55 PM on the due date listed above. Be sure to upload your files correctly the first time. If you have any problems prior to the submission deadline, please contact the instructor.

Extensions will not be granted for technology-related issues. Leave yourself enough time to complete the assignment, submit the assignment via MyGCC, and contact the instructor if you run into problems.